Anomaly detection in 5G beam propagation

Katarina Hellgren & Phiphi Tran
ka1527he-s@student.lu.se
ph5328tr-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisors:
Ove Edfors (LTH), Marcus Davidsson (Ericsson),
Thomas Svensson GE (Ericsson), Joel Bill (Ericsson)

Examiner:
Fredrik Rusek

May 31, 2021

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Advancements in today's technology has motivated invention of faster mobile communication systems. The fifth generation mobile network, 5G, is the latest version made by the third generation partnership project (3GPP) and expects to both increase connection speed and reduce latency, which eventually will make it applicable in supporting state-of-the art technologies such as virtual reality and self-driving vehicles among others. A major part of the 5G system is beam management, the system that controlls how signals or beams are assigned to user equipments such as phones and computers. Previous projects have looked at how machine learning could be used in order to improve beam management which has worked well. However, noise in the beams makes algorithms unreliable at times and signals unstable. The objective of this master's thesis was thus to apply machine learning in order to decrease the noise in the data, improving performance of beam management further. The machine learning algorithm that yielded the best results was a clustering algorithm called Density-based spatial clustering of applications with noise together with K-nearest-neighbour. The results implied that by applying the machine learning algorithms and hence avoiding noisy data, higher signal power, data transfere speed and lower control signalling overhead could be achieved.

Acknowledgements

We would like to give our thanks to our supervisor at LTH, Ove Edfors, for his help and support. In addition, we would like to thank Thomas Svensson, Joel Bill and Marcus Davidsson at Ericsson for giving us the knowledge and supervision needed to conduct this master's thesis. Lastly, we want to express our gratitude to Ericsson as a whole for giving us this opportunity.

Table of Contents

Acronyms

3GPP	Third Generation Partnership Project.
AI	Artificial Intelligence.
AoA	Angle Of Arrival.
BM	Beam Management.
BS	Base Station.
DBSCAN	Density-Based Spatial Clustering of Applications with Noise.
DT	Decision Tree.
iForest	Isolation Forest.
iTree	Isolation Tree.
LOF	Local Outlier Factor.
LOS	Line-of-sight.
MIMO	Multiple Input Multiple Output.
ML	Machine Learning.
mmWaves	Millimeter Waves.
NN	Nearest Neighbors.
NR	New Radio.
PBCH	Physical Broadcast Channel.
PSS	Primary Synchronization Signal.
RAT	Radio Access Technology.
RF	Random Forest.
RSRP	Reference Signal Received Power.
SSB	Synchronization Signal Block.
SSS	Secondary Synchronization Signal.
TA	Timing Advance.
TCP	Transmission Control Protocol.
TP	Throughput.
UE	User Equipment.

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Background and motivation

Advancements in today's technology has motivated development of faster mobile communication systems. The fifth-generation mobile network, 5G, is the latest generation by the third-generation partnership project (3GPP). It is expected to both increase connection speed and reduce latency, which eventually will make it applicable in supporting state-of-the art technologies such as virtual reality, self-driving vehicles, and remote surgery at hospitals to name a few [6]. A previous project at Ericsson [18] has suggested the use of different machine learning (ML) algorithms to improve the performance of 5G and 4G LTE. They have done this by improving beam management, i.e. how signals sent by the radio stations are allocated to different user equipments (UEs). However, they also found some room for improvements, one being that their algorithm sometimes chose beams that were side lobes to another main lobe, that is, leakage from another beam. This is because the side lobes at some locations have the highest signal strength compared to the other beams main lobe which makes it seem like the optimal choice. However, these side lobes are often unstable and causes non-optimal long-term results which the ML algorithm did not take into consideration. Hence, it is the objective of this master's thesis to apply an ML algorithm that identifies side lobes in order to prevent UEs from connecting to them when there are other, more stable choices, hence further increasing the beam management performance.

1.2 Problem formulation

The questions that is the objective of this thesis to answer are:

- Can we find an ML algorithm that classifies main and side lobes, respectively, with an accuracy above 70%?
- Can we reduce control signalling overheads, that is, the number of beam switches?
- Can we increase connectivity between the base stations and the UEs by preventing the UEs from connecting to those signals which are classified as side lobes by the ML algorithm?

1.3 Methodology

As a first step, a literature study related to beam forming and beam management is made to gain the right domain knowledge for the task, as well as a review of already attained ML knowledge. The following step is to attain familiarity with one of Ericsson's simulation tools partly by recreating similar results of previous work. The simulator is later on used to produce logs of data of a simulated reality of moving UEs and transmitted signals. Furthermore, some experimentation with different ML algorithms is pursued on the logs with a goal to first see whether an ML approach to solve the problem is applicable and thereafter decide which algorithm is the most suitable for the task. This is done by first training the ML algorithm on one part of the data log, called training set, followed by a prediction on the other part of the data, also called test set, in order to attain a relatively unbiased performance measure of the algorithm. The ML algorithm with the best results is then chosen as the best algorithm to solve this problem. All data together with the predictions is visualized to get a better understanding of what the algorithm is doing. As a last step, the predictions are used in the simulator to check whether control signaling is reduced, i.e. number of wide beam switches for a moving UE and whether the throughput is increased when avoiding the predicted outliers.

1.4 Previous work

The preceding work investigated how an ML method could be used to improve beam management for 5G New Radio (NR) to potentially optimize beam measurements and efficiently use that to find the best beam for single and multiple UEs. This would in theory yield stronger signals and better transmissions between the base station (BS) and UE. A feed forward neural network was used and did successfully perform the beam management measurements. For static environments, their algorithm had very high accuracy in terms of narrow beam prediction with a low average drop in signal strength power compared to the baseline. The baseline was the original performance without any algorithms applied. For moving environments, the algorithm was able to achieve a surprisingly high cell throughput growth for more connected UEs. Furthermore, for the high-speed environments, they were able to achieve slightly better throughput performance [18].

1.5 Delimitations

Due to the restriction in time, the scope needed to be limited. One delimitation is the investigated methods. These have been limited to cover commonly used, well-known and well-documented open source methods for outlier detection. This was considered the best option since it is the most time-efficient considering that we want to know whether we can find an ML algorithm that can classify beams correctly, and also check if it is better to avoid side lobes or not since this is currently unknown.

A second limitation is the simulation environment provided by Ericsson. Firstly, line-of-sight is assumed meaning that there are no reflections or objects blocking the signal as there normally are in the real world. As mentioned, it is currently not known if avoiding side lobes results in higher throughput and a start will therefore be to discover if ML algorithms are relevant to further explore by starting with ideal environments, hence line-of-sight. Secondly, the environment in the simulation is limited to one cell with a single base station. Handovers between multiple base stations for the user are therefore not considered.

Another delimitation is the method to verify the result and accuracy of the chosen methods. Since the simulator is not able to provide us with information about outliers, the labels were gathered from the clustering method thus limiting the best precision to the performance of this method. In other words, the ground truth had to be constructed artificially, making the ML algorithm of choice adopting the flaws that the ground truth might contain.

The final delimitation is the choice of number of parameters from the simulations. Due to the curse of dimensionality, few parameters have been chosen to maintain a high explainability and understanding. Visualizing the result has been important in this project to be able to create the ground truth and to confirm which ML algorithms are suitable and which are not.

Chapter 2
Wireless communication

In this chapter, theory with relevant abbreviations and concepts related to wireless communication are presented. The purpose is to give the reader the background needed on the subject to understand the motivations and choices made.

2.1 5G NR

5G NR is the latest radio access technology (RAT) developed by the 3GPP for the fifth-generation mobile network. The technology is based on the previous generation, 4G LTE, and comes with new and improved technologies such as millimeter waves (mmWaves), massive multiple input multiple output (MIMO) and beam management (BM). These changes will enable even larger data volumes, lower energy consumption, lower latency, and higher connection reliability, to mention a few [8].

2.2 mmWaves

The mmWave spectrum lies at frequencies between 30 and 300 GHz. This high range of frequencies offers a great advantage in bandwidth, leading to higher data speeds. Nevertheless, the signal has much worse propagation properties than those used in 4G LTE and suffers from relatively high path losses due to obstruction. The signal does also, in general, drop rapidly in strength when the relative positions between BS and UE increases, motivating the need of high directional connection between these components, i.e., high performing beam management [11].

2.3 Beamforming

Beamforming is a signal processing technique used for directional signal transmission and reception and is done by combining antenna array elements such that the signal is enhanced in some directions (constructive interference) and reduced in other (destructive interference) [8]. Beamforming in this context is used to direct the signals from the BS and UE towards each other such that information can be exchanged between them. Figure 2.1 illustrates a simplification of how the number of antenna elements can be combined to create different type of beams.

One antenna element corresponds to the beam radiating in all direction as for a lightbulb. With two antennas, it is possible for the beams to interfere constructively. Additional antennas will enable a more accurate direction for the beam to propagate in. More antennas will also yield side lobes as will be introduced below.

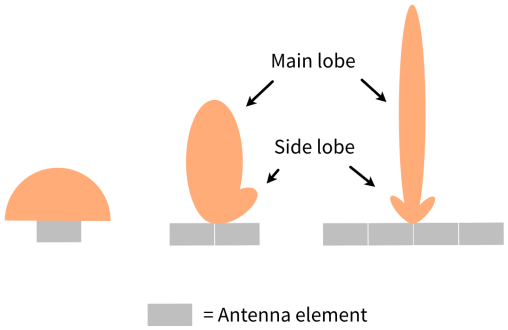


Figure 2.1: A simplified illustration of how beams can be created using different number of antenna elements. Above are beams shown for one, two and four antenna elements with respective main and side lobes.

2.3.1 Wide beams and Narrow beams

The antenna for our radio product contains 192 antenna elements as is portrayed in Figure 2.2. These antenna elements can together form either a wide beam or a narrow beam by using interference. The wide beam is a beam that covers a greater physical area where several UEs might be located while a narrow beam is a beam directed towards only one UE and is hence much more narrow. The Ericsson product works such that the wide beam with highest signal power is first chosen for a UE to connect to. This is also the wide beam which covers the area where the UE is located but sometimes there might be leaked signals from other beams that covers the same, which the UE might connect to instead if the signal power is higher. This is what we want to avoid and see if it leads to higher throughput. Furthermore, when the UE has connected to a wide beam, it will go through all narrow beams within the area of the wide beam and choose the narrow beam directed towards the UE, now with an increased signal strength and data transfer speed.

Within highband, analog beamforming is used which can only construct one beam at a time, in contrast to digital antennas which can construct several beams at a time. However, we will display the beams as if they were constructed simultaneously since the time intervals are significantly small.

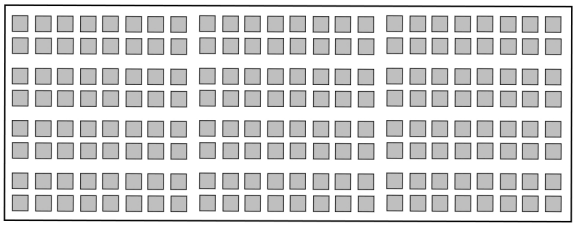


Figure 2.2: An overview of how the 8x24 antenna elements in our radio product are structured.

2.3.2 Main lobes and side lobes

In a directional antenna radiation pattern, where the greater power of the signal is being sent in specific directions, leakages of the signal often occurs due to inference, resulting in smaller beams on the sides of the original one. Hence, the beam can be divided into two categories: a main lobe and one or several side lobes. The main lobe is the lobe containing the higher power thus a greater field strength while the side lobes are the lobes in any direction other than of the main lobe. Any signal radiated in an unwanted direction is considered as noise to any other radio or UE, thus should be avoided [10]. Figure 2.3 demonstrates where one UE has been connected to a side lobe shown in grey.

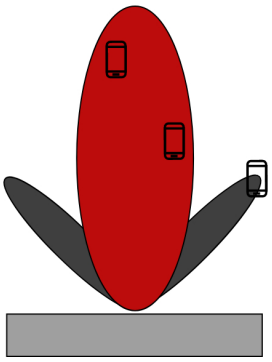


Figure 2.3: Demonstration of a main lobe and two side lobes where one UE has been connected to a side lobe.

2.4 Beam management

The procedure of BM is used to obtain the optimal connectivity of a beam pair between a receiving and transmitting beam generated from, for example, a UE or BS. In an environment free of obstacles called line-of-sight (LOS), as seen in Figure 2.4, the best beam pair would be the ones physically pointing towards each other. Due to the wavelength of the beams as well as environmental surroundings, the beams may be blocked, and a reflected beam can therefore result in a better connectivity [8]. Some constructive inference of leakage in beams could also result in side lobes with high signal power which might lead to better connectivity as well. In general, the procedure of BM can be divided into beam establishment, beam adjustment and beam recovery [8].

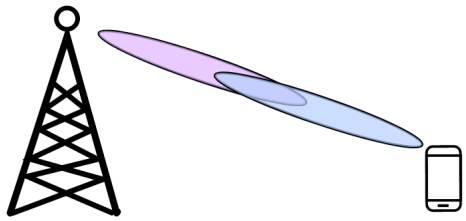


Figure 2.4: Connected signals between a BS and a UE.

2.4.1 Beam establishment

The first step is to establish connection between a UE and a wide beam covering the area where the UE is located. Thereafter, a *beam sweap* is performed. That is, we go through all narrow beams within the wide beam area and choose the one with the highest signal strength in the location of the UE. Then a beam pair in both transmission directions, downlink and uplink is established. That is, as in Figure 2.4, the BS sends out a downlink signal towards the UE and the UE sends an uplink signal towards the BS. When a connection between the beams is set, the beam pair is said to be initialized and one can assume that these beams will be used to transmit data [8].

2.4.2 Beam adjustment

After the establishment of an initial beam pair, reevaluations have to be done regularly due to the movements of the UE or changes in the environment. Beam adjustment can be further divided into reevaluation of the transmitter-side beam direction, given the receiver-side beam direction, and vice versa. This corresponds to downlink respectively uplink [8].

2.4.3 Beam recovery

Sometimes, a beam pair establishment is performed before the beam adjustment which leads to a beam-failure. The NR specifications include procedures to handle this, which is called beam-recovery and consists of the following steps: [8]

1. Beam-failure detection: the device detecting a beam-failure.
2. Candidate-beam identification: UE identifying a new optimal beam pair.
3. Recovery-request transmission: UE transmits a beam recovery request to the network.
4. Networks response to the beam-recovery request.

2.5 Parameters

The process of BM consists of several parameters covering complex concepts in wireless communication. Below the most relevant ones related to the thesis will be introduced.

2.5.1 Synchronization Signal Block

Synchronization signal block (SSB) refers to a block packed as a single block that always moves together. The procedure for a UE to acquire time and frequency synchronization with a cell is called cell search. During cell search operations the UE uses NR synchronization signals and the physical broadcast channel (PBCH) to get the necessary information required to access the cell. There are two types of synchronization signals: primary synchronization signal (PSS) and secondary synchronization signal (SSS) where the SSB consists of PSS, SSS and the PBCH. This signal block is sent by the BS and received by a UE which can then be used to measure the power of the received signal [2]. The theory behind the construction of SSB does not contribute much to the understanding of the main problem in the thesis. However, it gives a sense of the complexity of the theory behind the origin of the signals and was therefore considered worth mentioning without further elaboration. The time-frequency structure of an SSB is shown in Figure 2.5 for the interested reader.

2.5.2 Reference Signal Received Power

Reference signal received power (RSRP) is the measured power of the received signal obtained from SSB. UEs are assigned to the beam with the highest RSRP value, indicating the strongest signal. Sometimes, however, a side lobe can be the beam with the highest RSRP value resulting in an assignment to one of those instead, even though the signal strength over time might not be optimal due to the unstable properties of the side lobe [3]. RSRP is in our thesis measured in dB, thus minor changes can have a great impact due to its logarithmic scale.

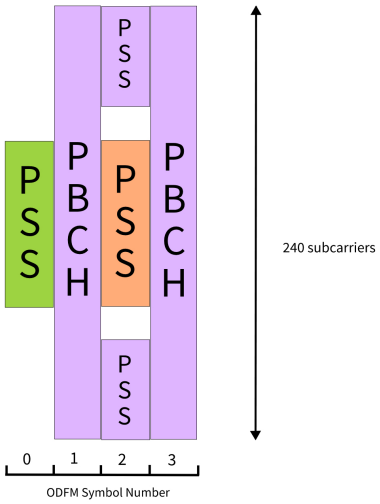


Figure 2.5: Time and frequency structure of an SSB

2.5.3 Throughput

Throughput (TP) is a measure of how many megabits per second (Mbps) that can be transmitted and received by a BS and a UE, respectively. It is measured in megabytes per second and hence tells us how fast data is being transmitted from the BS and recieved UE, or vice versa. TP is one of the most important parameters from a user's perspective since it is directly coupled to data transfer speed. Hence, higher TP always means faster data transfer for the user of the network. TP is regulated by the Transmission Control Protocol (TCP) which increases the data transmission speed until failure when it instead cut transmission speed off, creating a pattern similar to that in Figure 2.6.

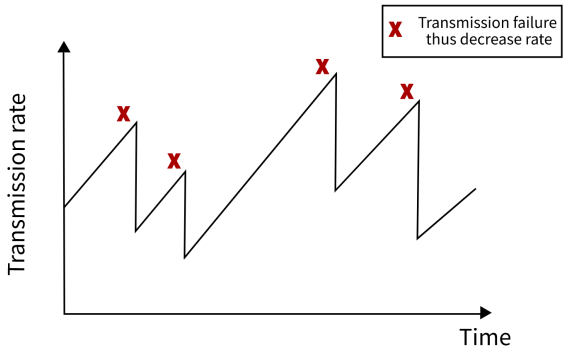


Figure 2.6: TCP regulating TP

2.5.4 Timing Advance

Timing advance (TA) is the control data sent by BS to UE which the UE uses to compensate for propagation delay in time, before the uplink signal is being sent back to BS [1]. A UE far from BS for example, encounters a higher propagation delay than a UE closer to the BS. Since the signal is traveling in the speed of light, it is possible to calculate an estimation of the distance between different UEs that later can be converted to the distance from the BS. This, together with the Angle of Arrival, described below, are relevant substitutes to the position of the UE since those locations does not exist in the real product and cannot be used other than for experiments. In the simulator, the distance which is supposed to be calculated from TA is instead replaced with the coordinates of the UE and an additional uncertainty of 9 m, to be as similar to an implementation of TA as possible.

2.5.5 Angle of Arrival

Angle of arrival (AoA) is the angle at which a signal arrives at the measuring station as seen in Figure 2.7. The AoA is estimated by antenna arrays and can be used as a part of estimating the locations of the UEs. AoA can be divided into two parts, the horizontal angle called azimuth, corresponding to the x- and y-axis in space, and a vertical angle called elevation corresponding to the z-direction [4]. In the thesis, only azimuth is relevant to locate the position of the UE in the cell since the distance will be known from TA. Elevation is indicating the distance from the BS and azimuth where on the corresponding circular arc the UE is located, hence only azimuth will be applied.

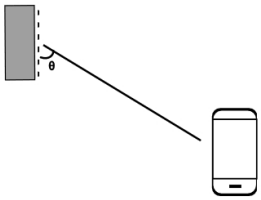


Figure 2.7: Angle of arrival, θ

Chapter 3
Anomaly detection

An anomaly, also referred to as an outlier, is a data point that differs significantly from the rest of the dataset. The deviation can indicate a variability in measurements, manual or equipment errors, contaminations, or frauds. Identifying and removing outliers from data is especially important when training ML algorithms for predictive modeling. Consequences of outliers can be skew statistical measures and data distributions, misleading representations of data and relationships [17], or higher energy consumption as in our case. For us, the side lobes are considered as the outliers and it is the objective to find an anomaly-detection algorithm that can identify these side lobes and eventually prevent the UE to connect to those. In this chapter we present and explain different types of anomaly detections and elaborate whether they might be useful for our specific problem.

3.1 Types of anomaly detection

Anomaly detection has become one of the most important issues in data mining and can be applied in a broad variety of real-world areas such as public health anomaly, intrusion detection, and credit card fraud. Within 5G, it has become harder to manage the large amount of data from constantly growing complex systems and two typical examples of an anomaly is spikes in base station load and, in a network, an increase in call drops [5]. There are many areas of application just as the number of algorithms [19]. Five categories of more commonly used anomaly detection techniques that will be investigated are statistical-, clustering-, distance-, density- and isolation-based.

3.2 Statistical-based

Statistical-based approaches generate a model describing the data distribution where statistical measures are applied, such as mean, variance and median. New unseen data points which does not fit the model, or have a low probability to fit, are considered as outliers [17]. Some traditional methods are based on standard deviation, histograms and boxplots which our data might be suitable to depending on the distribution within relevant features. For example, we could use a method that finds some statistical data on the RSRP, AoA and TA from the wide

beams and use that model to classify signals as either side lobes or main lobes by comparing the data with the statistical model.

3.3 Clustering-based

Clustering approaches follows the principle of treating data points far away as outliers. Far away in this context means data that has feature values that differs from what is the most common and the methods are usually based on either distance or density, dividing the dataset in different clusters depending on the similarity between the points. One can rather easily visualize a cluster of data points in the mind and conclude that outliers are those points which does not have many neighbouring points in close proximity and lies a bit outside the cluster of data.

By using the proximity between trained data, anomalous points can be identified. Distant clusters or clusters with small density is considered as anomalies. Clustering methods are not classifiers, thus prediction of potential outliers cannot be done using this approach [17].

For our specific problem, a clustering-based approach seem suitable due to the result from previous work [18] that has shown a formation of clusters within a cell belonging to respective wide beam. The cell for this example was visualized in two dimensions corresponding to the coordinates of a moving UE.

3.4 Distance-based

With a given distance measure, such as traditional Euclidean distance, data points with large deviation distance from a predefined number of nearby neighbors is classified as outliers. A relevant distance measure needs to carefully be chosen, for example depending on the dimension of the problem. An outlier in distance-based methods can also be characterized as having less neighbors within a fixed radius r. This requires computations of the distances between each sample in the dataset to gain prior knowledge about the dataset as a whole. There are several examples of techniques based on the traditional Nearest neighbor, where one of the most commonly used is k-nearest neighbor [17].

With a similar reasoning as for clustering-based approaches, a distance-based one might as well be suitable if a way to separate the outliers can be found and from there apply the method of comparing the number of neighbors with a close distance. This could be done both by using the coordinates of the UEs' or by using AoA and TA.

3.5 Density-based

The general idea behind density-based approaches is to compare the density of neighboring points with the density around its local neighbors. The relative density is compared to its neighbor to compute an outlier score, where the density around an outlier is significantly different to the density around its neighbors [7]. This

group of methods are adapted for local anomalies and, compared with distance-based methods, more efficient. Density-based approaches are also in general more efficient for data of high dimension, compared to distance-based which quickly can become complex [17].

A density-based approach might, for a similar reason as distance-based approach, be suitable if the outliers can be separated and from there compare the densities. Since the data will be of a higher dimension, a density-based approach seems to be more relevant than a distance-based one. For example, more correlations might be captured between the RSRP, TA and AoA when using a density-based method which could lead to more accurate results.

3.6 Isolation-based

Anomalies in the isolation-based approach are detected by isolating data points without relying on measures such as distance or density [12]. An example of an isolation process in two dimensions is to randomly draw vertical and horizontal lines among the data points and counting the number of lines that are needed to separate the data point from the rest. If the number of lines that is needed to isolate a data point is significantly smaller than average, it is considered as an outlier.

Using the properties of anomalies being the minority and consisting of few instances as well as these having different values from normal instances, it is therefore possible to separate the points and classify them as outliers. The main idea is that anomalies represent a small amount of the whole dataset, thus are likely to isolate quickly. Isolation based approaches are also able to adapt easier to data contexts because they do not have to perform computations of distances or densities, thus having a lower complexity and being more scalable [12][17]. With its advantages when it comes to adaption and low complexity, it is not surprising that this approach is commonly used and therefore considered highly relevant for our problem.

Chapter 4
Machine learning

Machine Learning (ML) is a subcategory of Artificial Intelligence (AI) and focuses on applications that learn from experience and improve their decision-making or predictive accuracy over time without being programmed to do so. Algorithms are trained on a large data set to find patterns that are used to make predictions or decisions on new data. ML methods can be categorized as supervised, unsupervised, or semi-supervised. In supervised ML, the labels of the training data (called *ground truth*) are known and can be exploited when creating the ML model by minimizing the *Objective function*, which measures the difference between our prediction and the ground truth. In contrast, when using unsupervised learning, there is no ground truth and the model is instead exploring and finding patterns and similarities in the given data. In unsupervised anomaly detection, any training data is not required and an assumption that normal data points are more frequent that abnormal ones is made. If the assumption is not true, the technique can suffer from high rate of false positives.

4.1 Decision Trees

A Decision Tree (DT) is an ML algorithm that uses classified data to make decisions based on specific attributes and have a tree-shaped structure, hence the name. The tree model is created by splitting the data with respect to some attribute and threshold, resulting in several new subsets of the original data set. These subsets are then split further with respect to some other attribute and continues in the same manner until there is either only one data point in a subset, when all belong to the same category, or if a predetermined depth or width of the tree has been reached. This model, with all the split attributes and thresholds is then used to predict on new data. All the points in the tree with the split attribute and threshold are called *nodes*, where the nodes at the bottom of the tree are called *leaf nodes* or just *leaves*. DT is a very intuitive ML model since it makes decisions in a similar way as humans do. Figure 4.1 shows an explanatory example of how DT make decisions, in this example deciding if walking or taking the bus is the best option. As can be observed, it is very similar to how humans make decisions and can be a powerful tool when incorporated into other algorithms. The blue nodes are the parent nodes and the yellow nodes are the leaf nodes [13]. DTs are not applied directly as a classifier in this thesis but exists as a part of other

algorithms that are used and has hence been included in the report.

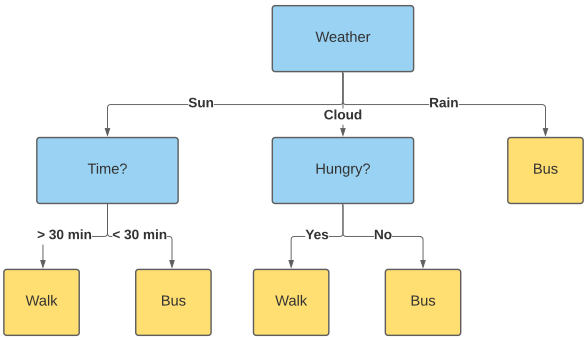


Figure 4.1: A clarifying flowchart of what a DT might look like with the example of trying to decide if one should walk or take the bus.

4.2 Random Forest

Random Forest (RF), or random decision forest, is a supervised ML algorithm consisting of several DTs. More specifically, the training data set is divided randomly, and each subset is sent into a DT. These DTs will have captured different patterns in the data so when making a prediction, the outputs of the DTs are incorporated to reach a final answer. This can be made either by majority voting, averaging or by using a weighted average of the outputs from the DTs. Combining several ML models, in this case many DTs, and incorporate their results for a decision or prediction is called an *ensemble* method [16]. An example of how RF can be divided is seen in Figure 4.2. As for the DTs, RF is not applied alone but is incorporated in a method explained below which has been tried out during this project and hence been deemed important to explain in more detail.

4.3 Isolation Forest

Isolation Forest (iForest) is a variant of a RF and is mainly used for anomaly detection. It works on the principle of isolating anomalies, in contrast to normal techniques that instead profiles normal points. iForest consists of an ensemble of Isolation Trees (iTree), which is the data that have been split into two sets with a random threshold. These two data sets are then further split into two new data sets each, creating a tree structure similar to Figure 4.1. A leaf node has been reached when it is possible to isolate a point, that is, divided a data set into two sets, one with only one data point (the isolated point) and one set with all the other points. We create many iTrees like this, hence creating an iForest, and

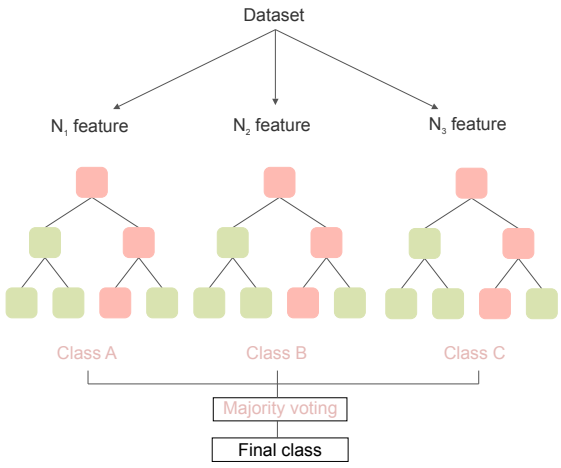


Figure 4.2: An example of a RF where the data set is divided into pure features where each one is sent in as training data for a small DT.

anomalies are the points that have shorter average path lengths in the iTrees since those points, on average, have been easier to isolate [12]. An explanatory image of the splits can be seen in Figure 4.3.

iForest is the common approach when using an isolation-based anomaly detection technique, as well as anomaly detection in general. With data in several dimensions combined with a familiarity working with the method in previous projects at Ericsson, iForest was a natural choice of method. In theory, data from a side lobe should be more easily separated because their parameters differ more to what is most common among all data points.

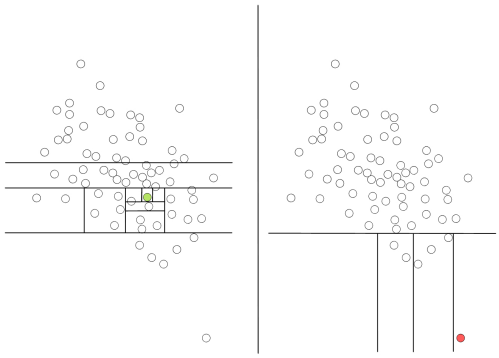


Figure 4.3: An overview of how the split in iForest can be made to isolate data points. To the right an outlier due to significantly fewer splits.

4.4 Density-based spatial clustering of applications with noise

Density-based spatial clustering of applications with noise (DBSCAN) is an unsupervised data clustering algorithm and, in contrast to other clustering algorithms such as k-means clustering, one does not in advance need to know the number of clusters the data is to be divided in. In addition, DBSCAN can find arbitrarily shaped clusters, has a notion of noise and is robust to outliers. The algorithm can be visualized in Figure 4.4 and abstracted accordingly:

- Find the points in the neighborhood within a radius of ϵ of each point and identify the core points with more than N neighbors, N being some positive integer. Both ϵ and N are to be determined in advance and hence a good domain knowledge is needed in order to make good such choices.
- Find the connected components of core points on the neighbour graph.
- Assign each non-core point to a nearby cluster if the cluster is an ϵ neighbour, otherwise assign it as noise [9].

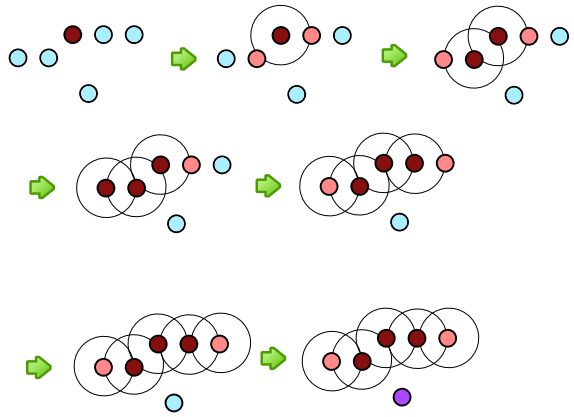


Figure 4.4: An explanatory image of the DBSCAN algorithm.

However, DBSCAN does not cluster data sets well if there are big differences in densities since the parameters, ϵ and N, cannot be chosen appropriately for all clusters. Nevertheless, DBSCAN does generally do a good job at clustering arbitrary shapes and figures [9], in contrast to more conventional clustering methods such as k-means clustering. An example of this is shown in Figure 4.5.

4.5 Local Outlier Factor

Local Outlier Factor (LOF) is an unsupervised method for anomaly detection measuring the local deviation of density of a given data point compared with its neighbors. The anomaly score of each data point is called LOF where the localness depends on how isolated the point is in respect to its k nearest neighbors. A point

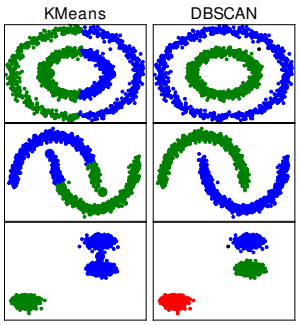


Figure 4.5: An example of what the clusters may look like when applying DBSCAN and k-means clustering, repsectively.

is considered as an outlier if it has a considerably lower local density of neighbors than its surrounding neighbors. LOF does also share some common concepts with DBSCAN such as core distance and reachability distance, used to estimate the local density. An advantage of LOF is the ability to identify a different type of outlier compared to traditional methods, such as identifying a point within a sparse cluster with similar distances to its neighbors as inliers, or a point at a relatively small distance to a very dense cluster as an outlier as seen in Figure 4.6 [7].

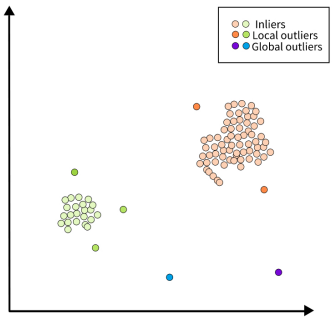


Figure 4.6: An example of the advantage of LOF being able to identify local and global outliers.

4.6 Nearest Neighbors

Nearest Neighbors (NN) is a range of methods based on finding a predefined number of trained data points that are closest in distance to the new point, and from

these predict a label. The standard metric measure is Euclidean distance but can in general be any. Since NN are simple methods remembering its data, they are usually considered as non-generalizing ML. Its simplicity is the foundation of many other learning methods and can provide functionality for supervised as well as unsupervised methods [15]. Figure 4.7 illustrates a simple example of how the method is classifying a data point.

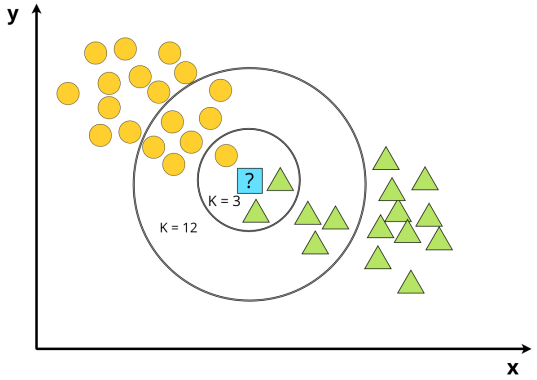


Figure 4.7: An example of how KNN chooses which class the sample belongs to. If K=3, the sample is a green triangle, if K=12 it will be a yellow circle.

4.7 Performance measures

When evaluating the performance of an ML model, there are many performance measurements that can be used. Some of the more common ones are *recall*, *precision*, and *F-score* that has been applied in this thesis and hence will be explained in more detail below. Before doing so, there are a few terms that need to be explained first:

- TP, in Section 4.7 only, is short for true positive. It is the total amount of points correctly predicted to be in a class
- TN is short for true negative and is the total amount of points correctly predicted as **not** being in a class
- FP is short for false positive and is the total amount of points falsely predicted to be in a class
- FN is short for false negative and is the total amount of points falsely predicted as **not** belonging to a class

More points classified as TP and TN, and less FP and FN, means a better ML model.

4.7.1 Recall and precision

The precision and recall are defined as follows and can be visualized in Figure 4.8 in order to get a better understanding of the terms.

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

To explain Figure 4.8 further, one can assume the data points in the left side of

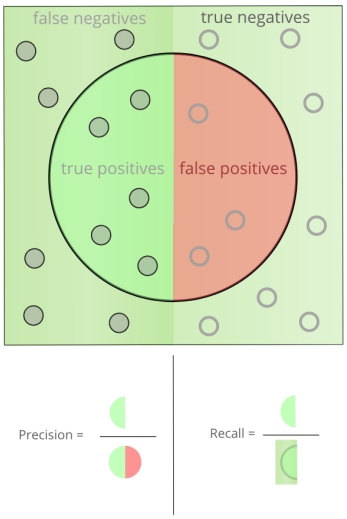


Figure 4.8: A visualisation of the terms recall and precision.

the image belonging to class A and the ones on the right belonging to class B. According to the figure, all points inside the circle have been predicted to belong to class A while the points outside of the circle have been predicted to belong to class B. Hence, the points on the left inside the circle are correctly classified as class A (TP), the points on the right side inside the circle are falsely classified as class A (FP), the points on the left outside of the circle are falsely classified as class B (FN) and the points on the right side outside the circle are correctly classified as class B.

In words, one can explain the recall as "of all the points that belongs to a certain class, how many of those did our model capture?" while precision can be expressed as "of all points that were predicted to belong to a certain class, how many of those were correct?". This means that if a model would predict all points to belong to a certain class, the model would get the highest recall score possible but probably a very low precision. On the other way around, if the model only predicted one point of many as belonging to a certain class and that prediction was correct, the model would get the highest precision score but a very low recall.

Optimally, one would want as high score as possible for both the recall and the precision. Nevertheless, for the objective of this thesis, the precision is deemed more important than the recall since the result would be much worse if an inlier was classified as an outlier than the other way around. If we for example accidently predict a side lobe to be a main lobe the UE will stay connected to that side lobe and the TP would be as good as it is in the currect product. However, if we instead accidently predicted a main lobe to be a side lobe, the result would be that a UE that already is connected to the correct beam with highest RSRP is connected to another beam instead. Connecting to an incorrect beam will probably result in lower TP since the surrounding UEs are indicating that there exist a more suitable choice of wide beam, at least for LOS. Hence, the precision is more important as a performance measure than the recall.

4.7.2 F-score

The F-score weighs in both the recall and precision in one value and is defined as

$$F\text{-}score = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

The constant β is used to weigh the F-score such that either the precision or the recall will have a greater effect on the result. A lower β means that precision weighs more than recall for example. We chose $\beta = 0.5$ as this is the most common when precision weighs more heavily than recall. As stated above, it is better to let UEs continuing staying connected to a side lobe rather than incorrectly classifying a main lobe as a side lobe, hence higher weight to precision [14].

Chapter 5
Method

Generally, when talking about methodology in ML applications, there are three core steps to consider. These are data collection, model training and model performance evaluation. This chapter will provide an overview of these steps as well as of the data and applied methods used to improve BM.

5.1 Data collection

As for any process including ML, the first part is to gather data which we obtained by generating simulations on a simulator Ericsson provided.

5.1.1 Data overview

The simulator tool could provide a variety of data from parameters when investigating BM. Relevant ones for the master's thesis are the following

- RSRP values for narrow/wide beam
- Coordinates of the UE
- Angle of arrival (AoA)
- Timing advance (TA)

where AoA and TA are used to calculate the position of the UE. User position is accessible in the simulator but is not known in reality and will not be applicable in the final result. The coordinates are, however, used to identify the number of outliers, which are used to generate a baseline to compare the accuracy of other methods, using existing parameters in the real product. This will further be elaborated in Section 5.2.

5.1.2 Simulation tool overview

Environment

The simulated data traffic used in this master's thesis has been generated by a simulator tool developed internally by Ericsson, providing ideal 5G NR environment and supporting features such as mmWave frequencies, beamforming and BM.

Real scenarios are recreated to enable incorporation with algorithms that might improve the final product. In this thesis, the simulation tool is used to discover if implementation of anomaly detection methods can improve BM. The generated data is stored in log files where values of each parameter is post-processed in Python.

Simulation parameters

- Simulation time: 1000 s
- Carrier frequency: 28 GHz
- Carrier bandwidth: 200 MHz
- Cell shape: Hexagon
- Cell radius: 150 m
- Number of wide beams: 12
- Number of narrow beams: 12 per wide beam
- Antenna
    - Height: 25 m
    - Zenith angle: 23°
- UE
    - Movement: Random mover
    - Moving speed: 1.0 m/s

where the zenit angle for an antenna describes how much it is tilted relative to its vertical axis.

When a simulation starts, UEs will randomly move over an area of a cell, shaped as a hexagon as in Figure 5.1. One UE at a time will randomly be placed to move around nearby for a limited time before it is being replaced by a new UE at another position in the cell. In our simulations, the information of the UE was collected approximately 10 times during its lifetime. The movements are therefore not based on reality but statistics instead. UEs will move around while trying to establish a connection for data transmission with the BS. Figure 5.1 illustrates the position of the base station in the hexagon cell as well as where the propagation of different beams originate.

The hexagon has a radius of 150 m, measured from the center to each of its six corners. Once initialized, the UE will normally be connected to the wide beam transmitting the highest RSRP value and thereafter adjustments will be made following the movement of the UE. Using the approach with ML will, before a UE has been initialized, find the most suitable wide beam to connect to based on a trained model. More precisely, if the UE is connected to a side lobe, the ML method will be able to detect the inefficient connection and change to the optimal one based on particular condition elaborated in Section 5.6.

In this thesis, LOS is used, hence the clean pattern with visually well-defined borders as will be seen in upcoming section. Including obstructions of the beams would with high probability result in significantly more outliers but will not be covered here, where focus is on LOS.

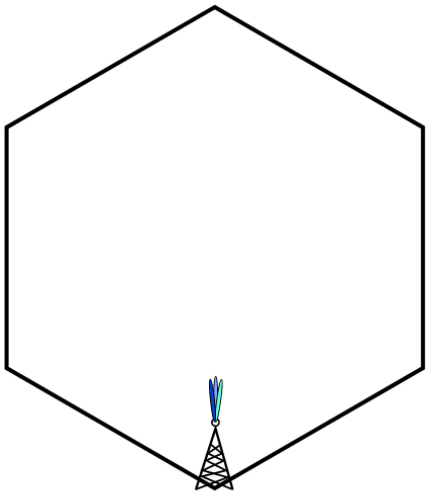


Figure 5.1: The hexagon-shaped cell UEs are randomly moving across during simulation.

Python

The ML algorithms are implemented in Python, both for training and prediction, enabling the usage of open source libraries such as scikit-learn containing simple and efficient ML tools. The postprocessing of data is also made in Python.

5.2 Understanding the data

The first step was to understand and visualize the distribution of wide beams in the hexagon cell. In Figure 5.2 one can see the distribution of wide beams based on the x- and y-positions of the UE in the cell, both in two and three dimensions. In 3D, the third dimension represents the connected wide beam index. Outliers can in 2D be seen as a sample of mismatching color with respect to the surrounding area while they in 3D are scattered around the clusters. Notice that only nine, out of twelve existing, wide beams fit the cell in our scenarios and for the specific radio product. In Figure 5.3, one can see all the samples where the UEs have been connected to one certain wide beam, here the turquoise colored one corresponding

to wide beam 8. How the beams propagate and scatter from the antenna have been visualized in Figure 5.4. From the same figure, the wide beam with index 1 correponds to the dark red area, index 2 to the light red area, 3 to dark orange, 4 to lighter orange and so on for the remaining wide beams.

Coordinates of UEs have been used for an easier understanding but, as mentioned above, do not exist in the real product and have in Figure 5.5 been replaced by the parameters AoA and distance. Coordinates are used as an analogy where the figures with AoA and distance shows the correspondence.

One might wonder how these outliers occur and how they can exist in Ericsson's product. There is currently not a clear explanation for the outliers but what we have observed in the simulated environment for the specific radio is that the side lobes at some locations reaches an energy level to be considered as the strongest. This is what we call an outlier. The strength of RSRP of the side lobes is depended on the configuration of beams in the radio which probably is a trade-off that is accurately calculated in the product. There are furthermore some observations that we can observe and discuss.

The connected wide beam index represents the wide beam with the highest RSRP the UE at that specific position is receiving. Figure 5.7 shows the strength of the RSRP for every sample for a specific wide beam with red as strongest and blue as weakest. Figure 5.7a corresponds to wide beam 4 with the orange colored area as in Figure 5.4, Figure 5.7b to wide beam 5 corresponding to yellow, Figure 5.7c to wide beam 7 corresponding to turquoise and Figure 5.7d to wide beam 8 corresponding to light blue.

From Figure 5.7, one can see that the signal for a specific beam is the strongest in their respective area and fades with the distance, more to the sides than in the direction the beam is propagating in. It is also possible to see that the coverage area of the upper beams, such as wide beam 7 and 8, are greater due the zenith angle. The farther the beam hits the ground, the broader area it will cover as seen in Figure 5.6

One can also notice that the center wide beams, here 5 and 8, seem to have some strong signals along a line on each side. Some of those samples along these lines are also the ones visible as outliers. It is unclear if this actually occur in reality, but does in our environment with the specific beam weights according to the radio product we have been given. The signals along the line are the side lobes, visualized in Figure 5.8. These side lobes have lower RSRP than their main lobe, but in some cases still higher than the main lobe for the corresponding area, thus an outlier.

Another reason for the occurrence of outliers can be that the RSRP value is not only dependent on the energy sent by the BS, but also the UEs receiver beam. One factor that can affect the measured RSRP in UE is how the UE is set to handle its receiving antennas. That might affect how a main lobe and an overlapped side lobe is being received, if they do not have the exact same angle for example. The UE has antennas it is directing towards the direction where it receives the highest RSRP for each wide beam, which should correspond to the same direction as the BS. How much this factor is supported in Ericsson's simulator is unclear to us.

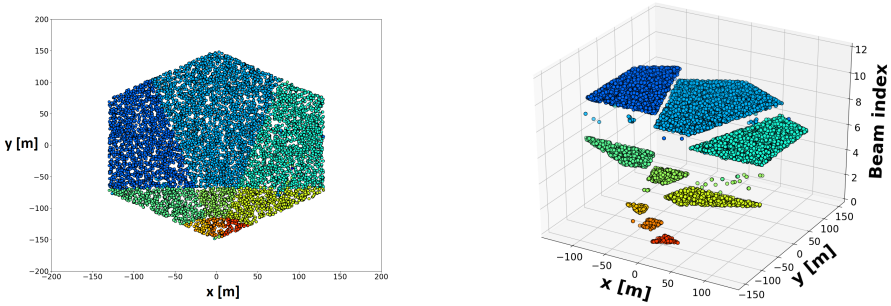


Figure 5.2: The distribution of connected beam with respect to the position in 2D and 3D in a cell.

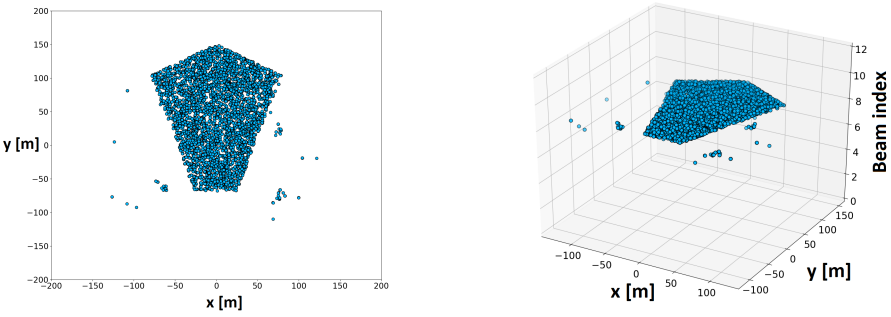


Figure 5.3: The distribution of connected UEs to wide beam 8 in 2D and 3D.

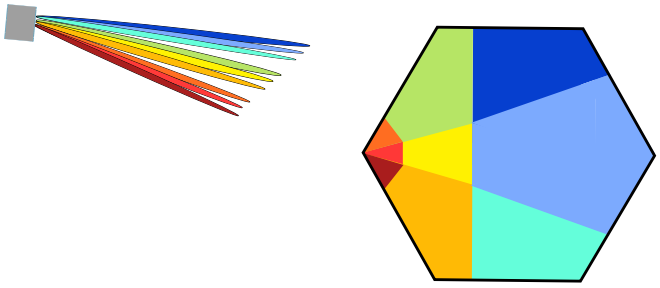


Figure 5.4: A conceptual overview of the distribution of wide beams from antenna to cell.

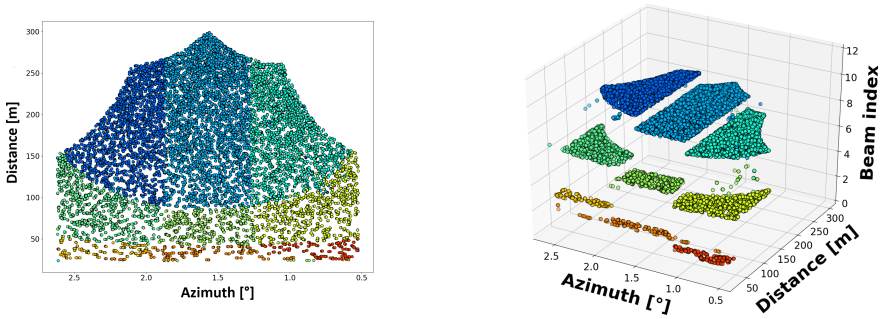


Figure 5.5: The distribution of connected beam with respect to the position in 2D and 3D in a cell using AoA and distance.

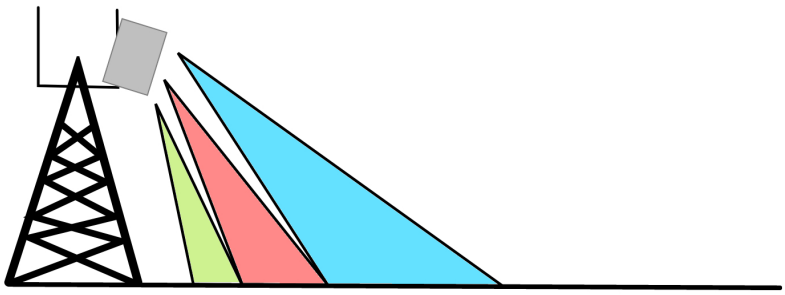


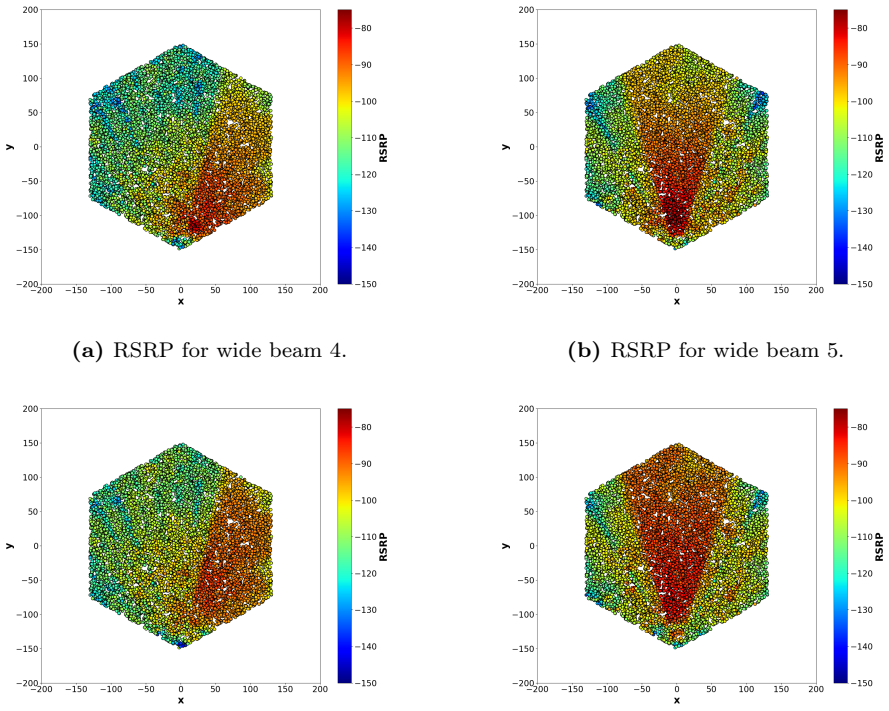
Figure 5.6: Illustration of the coverage area depending on the distance from the antenna and where the beam hits the ground.

5.3 Training

After the data had been generated in a LOS-environment, the proceeding step was to implement different ML methods to apply the data on. This would be done before creating models which later could be evaluated on new data that the model had not encountered before. Both the training and test sets contained roughly 25000 samples each where approximately 150 of those were considered to be outliers.

iForest

For the isolation-based approach, iForest was chosen since it is one of the more generally sophisticated anomaly detection algorithms and has been used for similar problems before in projects at Ericsson. The hope is that, based on RSRP, AoA



(a) RSRP for wide beam 4.

(b) RSRP for wide beam 5.

(c) RSRP for wide beam 7.

(d) RSRP for wide beam 8.

Figure 5.7: The strength of RSRP in different wide beams.

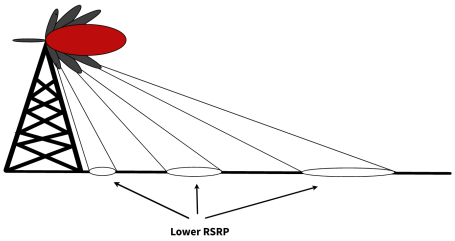


Figure 5.8: The propagation of side lobes with lower RSRP than the one for the main lobe.

and TA in an LOS-environment, the algorithm will be able to detect when a UE is trying to connect to a side lobe. iForest should be able to identify the side lobes quite well due to the difference in at least one parameter.

Hyperparameters such as number of base estimators in the ensemble, batch size in the estimator and proportion of outliers in the data set were tuned to optimize the method. iForest classifies the samples as either inlier or outlier that could be visualized.

DBSCAN

Another method commonly used for outlier detection is DBSCAN. However, the algorithm is fundamentally different compared to iForest since it is based on clustering, more similar to methods like k-means clustering or KNN to name a few. Nevertheless, it might as well be a useful method to apply since we have clusters of beams, meaning that DBSCAN might be able to cluster the main lobes and leave the side lobes out as outliers. DBSCAN is, however, a clustering method and cannot be used as a classifier. Our solution has therefore been to use the generated labels from DBSCAN to train a simple KNN working as its classifier. The labels returned consisted of -1 for outliers and otherwise, in our case, a number between 1-9 representing the cluster the sample belongs to. Some samples from wide beam 10, 11 and 12 exist but does not fit the cell and were classified as outliers.

A hyperparameter tuning was first done manually showing that the key parameters were the maximum distance between two samples to be considered as neighbors, ϵ, and the minimum number of samples in a neighborhood to be called a core point, N. Working in three dimensions was crucial for us to verify the correctness of the methods. Starting with coordinates, in x and y, and wide beam index, the radius ϵ could specifically be set to a predefined distance. The distances between neighboring samples were relatively small when running the simulator for a relative long period and ϵ was therefore considered reasonable to predefine.

Scaling the third dimension which consisted of the wide beam indices, with a factor of 10, separated the correct wide beam clusters from each other, as well as making the outliers more distinct. The radius ϵ could therefore be set to slightly under the scaling factor to ensure the distance from neighboring clusters always being less than the scaling factor. Separating the clusters from each other will also separate the outliers from the clusters.

The parameter N, can be set to one single neighbor to easily detect separate outliers. It is, however, not suitable when there exist clusters of outliers. In our simulation, approximately 10 close positions are registererd during the life-time of a UE. This parameter had to be adjusted depending on the sizes of outlying clusters, but due to the ability of clustering of DBSCAN the problem could easily be solved by increasing N until the desired number of separate clusters had been achieved. The advantage with DBSCAN is that the method also returns labels on which cluster the samples belong to, outliers counted as one common cluster. In a cell with nine wide beams, the number of neighbors N could be adjusted until ten clusters had been reached. This can automatically be done in an environment with LOS on the training model to ensure the best result. The number of outliers in a cluster varies depending on seed and scenario and did for example in one of our simulations consist of 20 samples. To make it more general, N is required to be relatively high to cover the number of outliers in an outlier cluster, but still not as high as the total number of clusters for one of the wide beams. Figure 5.9 shows the distribution of samples for each wide beam in one data set where the number of wide beam 8 is dramatically higher than the others.

Replacing coordinates with AoA and distance, the similar approach as above could be used. Due to the small difference in angle, ranging from 0.5 to 2.5 degrees in total, this parameter had to be scaled to be in a similar range as the coordinates.

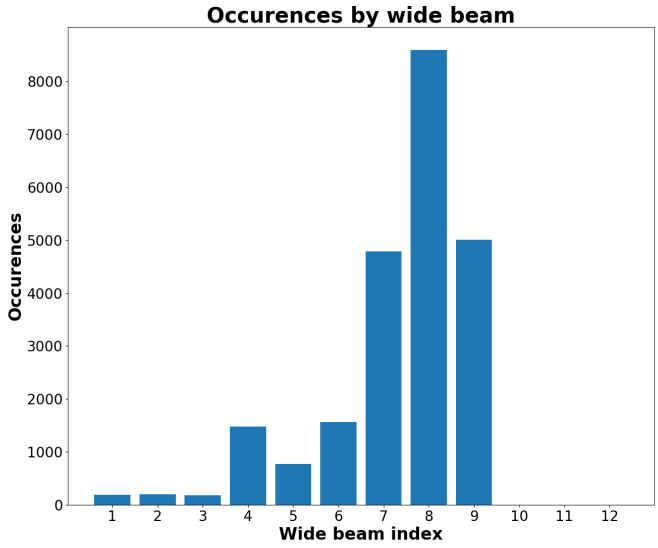


Figure 5.9: An example of how the wide beam distribution of the data points could be.

The scaling is however the same for all scenarios and ML methods if the radius of the cell stays consistent.

LOF

Since we can see LOF as a less sophisticated version of DBSCAN, we should also suspect the method to work at most as well as DBSCAN but not better, assuming we have chosen all parameters optimally. This is due to that LOF probably will identify a dense area of outliers as a cluster. Unlike DBSCAN, LOF is not a clustering method and has a built-in KNN that can be used as a classifier. Hence there is no need to incorporate a separate classification method as for DBSCAN.

The relevant hyperparameter to tune ended up being the number of neighbors only. More hyperparameters exist but did not have a significant impact, as the number of neighbors to take in consideration did. Finding a suitable value was a more challenge since LOF labels the data points as an inlier or outlier only. Similar method as with DBSCAN was not applicable and to have a comparable result in accuracy, the number of neighbors generated from DBSCAN was applied here.

KNN

We want a method that can identify outliers in beams and since these data points are distributed in space, a distance-based method such as KNN would be interesting to investigate further. Ideally, the algorithm would be able to classify points

closer together, i.e. that lies within the main lobe, as inliers and points further away as outliers based on different distance metrics available.

Since we are in a three-dimensional space, Euclidean distance is still possible to apply to the problem. Other distance metric that could be relevant in a higher dimension are for example Mahalanobis distance, taking the covariance between the features in consideration. As can be understood in the name of KNN, the number of neighbors is considered as a crucial hyperparameter. Similar as with LOF, a number of neighbors could not automatically be generated and the same number as for DBSCAN was applied.

Classification

All of the above methods, except iForest, uses KNN to classify. What KNN does is to return the indices of its nearest neighbors from training data. The total distance could be calculated where the scaling factor was divided with the number of nearest neighbors to use as threshold. With a scaling factor of 10 and using three nearest neighbors in prediction, an average distance for every sample could be generated in a chart as seen in Figure 5.10. A threshold of roughly three could then be used where all samples above the threshold were considered as outliers while those under were considered as inliers.

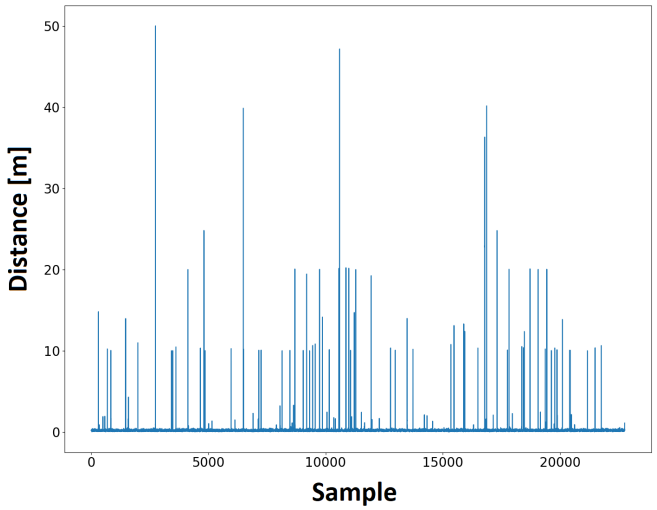


Figure 5.10: An example of how the average distance from each sample and its, in this case, three nearest neighbors were. The threshold would here be around 3.

5.4 Baseline algorithm

To validate the results, a baseline is required consisting of labels of each sample that indicates if it is an outlier or not. We did, however, not have any labels to compare with and did therefore need to create our own baseline. This was done by using the result from the best method of detecting outliers using the coordinates of the UEs' position. The method considered as the best was the method being able to visually identify the greatest number of outliers. The baseline was only used for evaluation to get a value of how well our algorithm was performing. The indices from the result in the baseline was compared with the labeling from the result using our chosen ML algorithm to generate performance measurements. Why using the indices generated from positions were considered more accurate is due to the evenly spaced samples in the cell, since the UEs' are moving randomly over the cell in the simulation. For a long simulation run time, the randomly movements should statistically cover the whole area.

The corresponding pattern using AoA and TA are not as equally divided which can be seen in the density of the separate clusters in Figure 5.2 and Figure 5.5.

5.5 Testing performance

The third and last step when using ML is to test it's performance. This was done by letting the method predict on new data, i.e. data that has not been used when training the model. The predictions made by the ML model are compared to the baseline that contains the labels from the training with coordinates. This comparison was made by calculating the precision, recall and F-score of the model, giving a value on how well the model performs. It is more important that the data points we predict as outliers actually are outliers than to capture as many of the outliers as possible. This is because we believe it would do more harm if a UE happened to be misclassified as being a part of a side lobe when it actually is in a main lobe. It would result in two unnecessary beam switches, one to connect to the the side lobe and another one to disconnect, yielding in an overall lower TP due to the side lobe and increased usage of energy for the beam switch. With this reasoning, we value precision higher than recall and has hence chosen the β-value as $\frac{1}{2}$ to incorporate that fact.

5.6 Beam update after prediction

After deciding upon the optimal ML model to use for prediction, the results were applied to decide a more suitable choice of wide beam.

This was done by using the classified outliers, assign them the wide beam index with the second highest RSRP and run the prediction model on the updated data. Thereafter, the procedure was repeated for the remaining outliers and with the wide beam with the third highest RSRP. If there still existed outliers, then these would keep the wide beam index received from the simulator. One can see it as a trade-off between a great connection based on wide beam RSRP and total energy consumption.

The next step was to store the updated wide beam indicies from predictions by the ML model, which would be used for verification of connectivity improvement. To verify that switching wide beam is a preferable choice to improve connectivity, one can look at highest RSRP for the narrow beam in the current wide beam and compare that with the highest RSRP for the narrow beam in the updated wide beam. A higher RSRP compared to the best narrow beam in the reference wide beam will indicate an improved connectivity. This had to be our approach to measure connectivity due to difficulties in the simulator. By using the exact same seed in the simulator, the movements and information of the UE will be conserved and thus the connections will be made in the same order. When a UE was at a time slot classified as an outlier, changes in the simulator code would force a connection to the predicted wide beam instead. The predicted wide beams were read into the simulator through an external file.

To verify that fewer beam switches were made and thus a lower energy consumption, they had to be counted. Samples in each area covering a wide beam were extracted, where the same random order of connection from the simulation were preserved. Doing so would also preserve the random distribution as used in the similator. One could also make a more realistic approach by counting the switches as if the UE walked in a continuously manner by sorting after position. For future implementation in the simulator, the first approach was however preferred. Thereafter, the total number of beam switches were counted before the ML model was run and modifying the wide beam indices of the outliers. Beam switches were again counted and compared to the first sum.

In addition, as a last step we wanted to look at the TP explicitly to verify if we could see a general improvement there as well. This was done by letting a UE stay put at a predefined location, which the model had classified as an outlier, while measuring the TP both in downlink and uplink with the wide beam suggested by the simulator and the predicted wide beam. The average of the two TP measurements were compared to verify any trends.

Chapter 6
Results

This chapter presents results from simulations and their performance with respect to their respective baseline algorithm, as in in Figure 5.2. Firstly, visual results from the methods will be presented to get an overview of the precision of outlier identification. Thereafter, the performance based on AoA and distance will be compared with baseline generated from position.

6.1 Clustering with DBSCAN

The result from clustering using DBSCAN based on position can be seen in Figure 6.1 while Figure 6.2 shows the clustering using AoA and distance. The left figure presents samples classified as inliers, color-coded to which beam the UEs' are connected to, while the right consist of all samples where classified inliers are colored as green and outliers as red. As one can see, DBSCAN is successfully able to separate the outliers when using positions and present visually clean clusters. With AoA and distance, the most clear outliers are visually identified.

6.2 Local Outlier Factor

The result from LOF based on position can be seen in Figure 6.3 while Figure 6.4 shows the clustering using AoA and distance. The left figure presents samples classified as inliers, color-coded to which beam the UEs' are connected to, while the right consist of all samples were classified inliers are colored as green and outliers as red. LOF seem to visually identify as good as DBSCAN, with the exception of a few local dense clusters such as the ones between wide beam 6 and 9, and wide beam 8 and 9.

6.3 Nearest Neighbor

The result for NN, in Figure 6.5 and Figure 6.6, is presented in a similar way as for DBSCAN and LOF. As one can see, NN is significantly worse than the two methods above. The scattered clusters, for example between wide beam 4 and 7 are not counted as outliers.

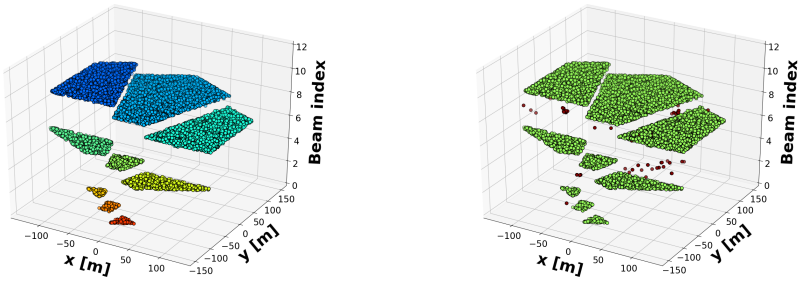


Figure 6.1: The left figure shows the data of inliers when DBSCAN has been applied while the figure to the right shows the corresponding predicted inliers in green and outliers in red. Both figures using x- and y-coordinates of the UE.

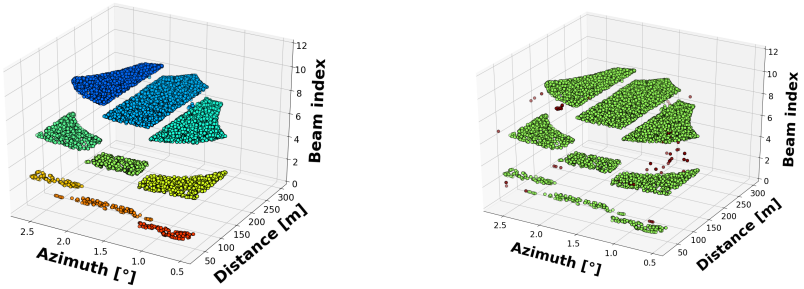


Figure 6.2: The left figure shows the data of inliers when DBSCAN has been applied while the figure to the right shows the corresponding predicted inliers in green and outliers in red. Both figures using AoA and distance from TA.

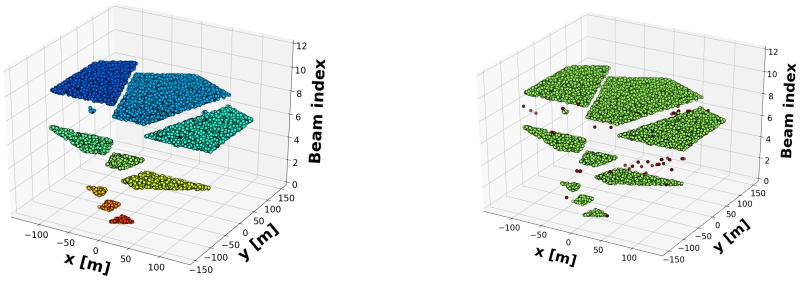


Figure 6.3: The left figure shows the data of inliers when LOF has been applied while the figure to the right shows the corresponding predicted inliers in green and outliers in red. Both figures using x- and y-coordinates of the UE.

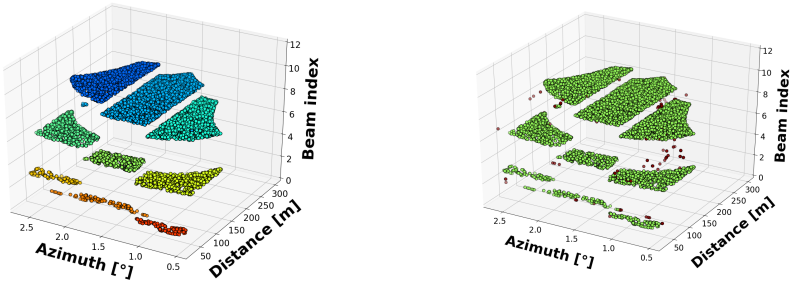


Figure 6.4: The left figure shows the data of inliers when LOF has been applied while the figure to the right shows the corresponding predicted inliers in green and outliers in red. Both figures using AoA and distance from TA.

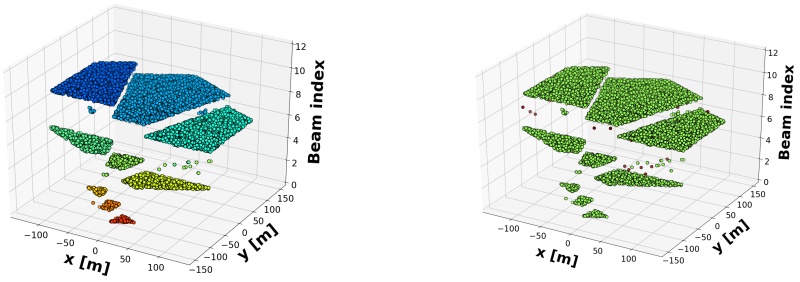


Figure 6.5: The left figure shows the data of inliers when KNN has been applied while the figure to the right shows the corresponding predicted inliers in green and outliers in red. Both figures using x- and y-coordinates of the UE.

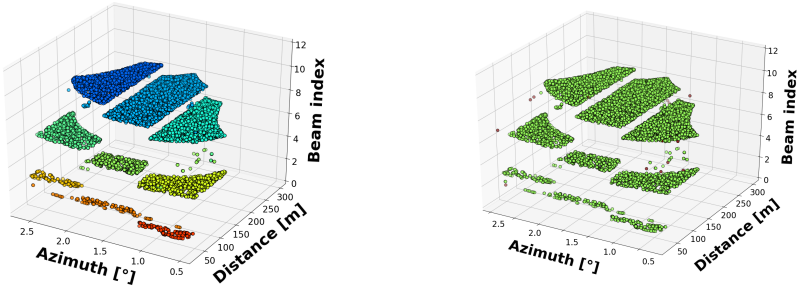


Figure 6.6: The left figure shows the data of inliers when KNN has been applied while the figure to the right shows the corresponding predicted inliers in green and outliers in red. Both figures using AoA and distance from TA.

6.4 Performance measures

The performance measures from above methods based on their ability to identify outliers are presented in tables. Table 6.1 presents the performance measures based on coordinates while Table 6.2 based on AoA and distance. The third table, Table 6.3 presents the result from different data sets using DBSCAN based on their respective baseline. A new baseline was generated for each dataset in a similar manner as described in method.

Table 6.1: Performance measures for different ML methods based on coordinates, compared to baseline generated by DBSCAN. For F-score, precision is weighted higher than recall.

Method	Precision	Recall	F-score
DBSCAN	1	0.993	0.999
LOF	0.699	0.623	0.683
KNN	0.958	0.167	0.491

Table 6.2: Performance measures for different ML methods based on AoA and distance using baseline generated with DBSCAN based on coordinates. For F-score, precision is weighted higher than recall.

Method	Precision	Recall	F-score
DBSCAN	0.952	0.855	0.931
LOF	0.688	0.768	0.703
KNN	0.950	0.138	0.436

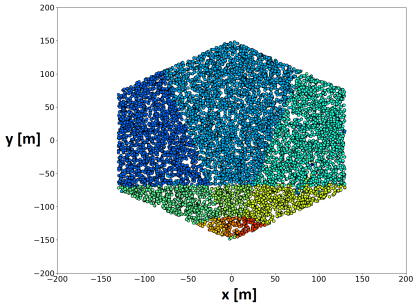
Table 6.3: Performance measures for different data sets based on AoA and distance using using a baseline generated with DBSCAN for each set respectively.

Method	Precision	Recall	F-score
Dataset 1	0.952	0.855	0.931
Dataset 2	0.973	0.775	0.926
Dataset 3	0.973	0.783	0.928
Average	0.971	0.804	0.932

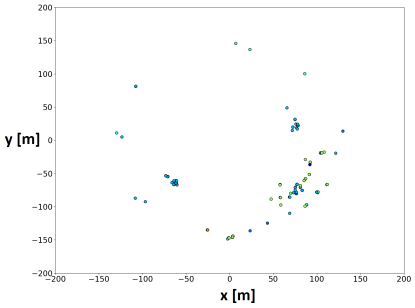
6.5 Beam update after prediction

The best method, DBSCAN, was then applied to see if the identified outliers had an inlier among of the other wide beams with higher RSRP. Figure 6.7 shows how

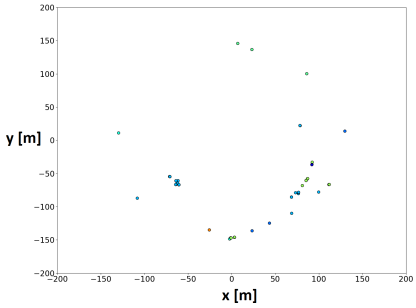
the number of outliers in two dimensions are reduced by applying the method on those connections classified as outliers, with the upcoming beam with highest RSRP. As an example from one of the data sets as seen in Figure 6.7a, Figure 6.7b shows 138 outliers from wide beam with the highest RSRP, Figure 6.7c the remaining 53 outliers from wide beam with second highest RSRP and Figure 6.7d consist of the 15 remaining outliers from the third highest RSRP.



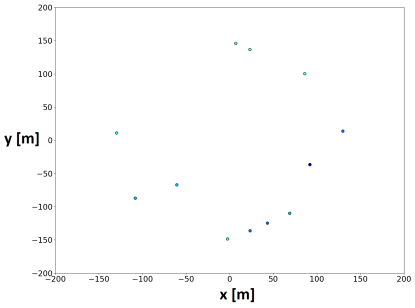
(a) All samples in the dataset.
.



(b) 138 outliers from the wide beam with highest RSRP.



(c) 53 remaining outliers from the wide beam with second highest RSRP.



(d) 15 remaining outliers from the wide beam with third highest RSRP.

Figure 6.7: Improvements when applying the ML method can be seen by the reduced number of outliers.

When investigating the total number of outliers in each wide beam, the distribution could look like as in Figure 6.8. The figure shows the distribution of the outliers in Figure 6.7b.

6.5.1 Wide beam switches

The simulatioin was run again with the exact same seed and now forced to choose the model's predicted wide beam for the classified outliers. Figure 6.9 shows five examples of the number of beam switches made both before and after our method had been applied, hence avoiding making the UE connect to side lobes.

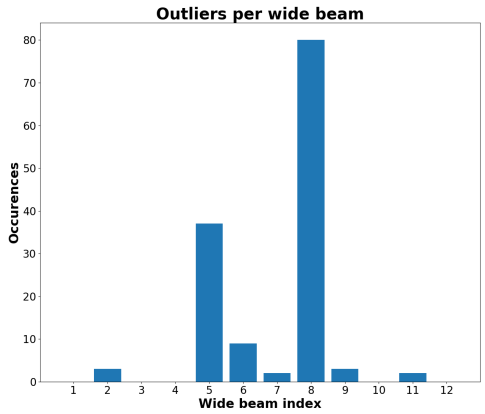


Figure 6.8: The distribution of which wide beam the outliers, in Figure 6.7b, belong to.

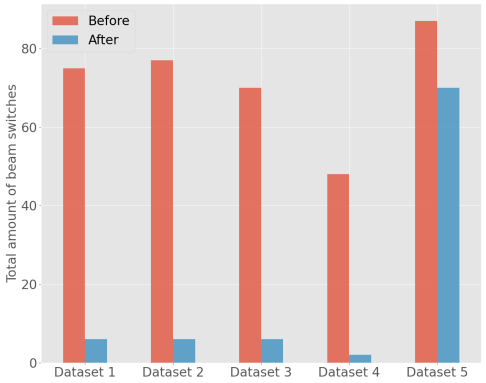


Figure 6.9: Number of wide beam switches before and after the outliers were discarded.

6.5.2 Wide beam and narrow beam RSRP

Some examples of the consecutive wide beam inlier search, when forcing the choice of wide beam for classified outliers, and change in the narrow beam RSRP can be seen in Table 6.4. For the first example, the simulator had chosen wide beam 6 which was classified as an outlier. The wide beam with next highest RSRP was 9 which also was classified as an outlier. The third highest RSRP belonged to wide beam 2 which our model classified as an inlier. Note that more samples were investigated and that the three data points in the table are the mean value of 10 samples respectively.

Table 6.4: Examples on some consecutive searches for wide beam inlier.

Wide beam predictions	Old RSRP	New RSRP
$6 \rightarrow 9 \rightarrow 2$	-95.5948	-114.7295
$5 \rightarrow 4$	-99.9155	-95.2112
$8 \rightarrow 6$	-104.5389	-103.6732

On average when using the predictions from our model, the wide beam RSRP decreased with 1.3 % as seen in Figure 6.10. At the same time, when only one beam prediction was needed, the average improvement of narrow beam RSRP was 3.0 % better as seen in Figure 6.11. Both are visualized in a boxplot showing the percentage difference in RSRP, now choosing the wide beam with the second highest RSRP.

Because of complications to generate narrow beam RSRP belonging to predicted wide beams, only 10 different UEs were used for the result. One UE is registered roughly 10 times during its life-time and will make minimal movements thus similar values for position and RSRP.

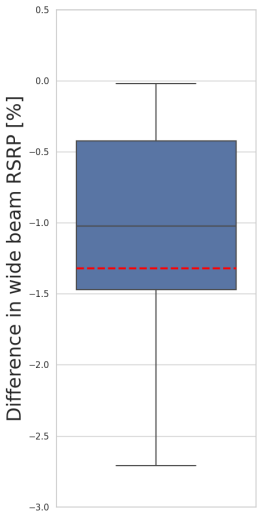


Figure 6.10: A boxplot of the difference in percentage of wide beam RSRP after a beam switch. The red dotted line marks the average.

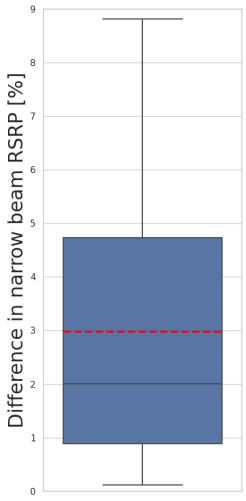


Figure 6.11: A boxplot of the difference in percentage of narrow beam RSRP after a beam switch. The red dotted line marks the average.

6.5.3 Throughput

Forcing the simulator to choose the model's predicted wide beam for those samples classified as outliers resulted in an average increase in TP. Figure 6.12 shows three examples of the average TP for a single UE using TCP over 30 s for three different datasets, both before and after we applied our ML algorithm and avoided the side lobes. Shorter simulations of 5 seconds were performed for the remaining samples of the investigated ones for narrow beam RSRP above.

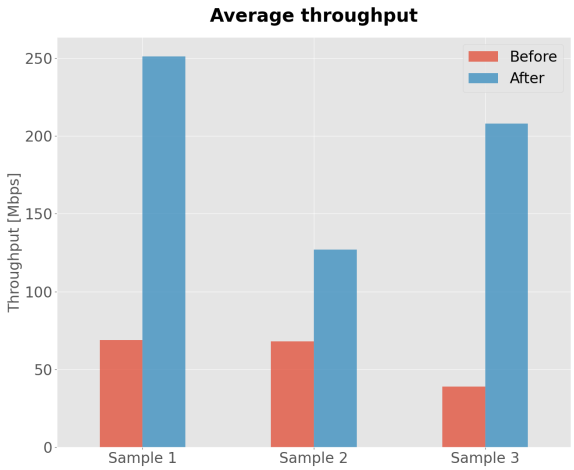


Figure 6.12: Average TP for a single UE using TCP over 30 s for three different data samples, both before and after side lobes were avoided.

Chapter 7
Discussion

In this chapter, the results from previous chapter will be discussed to build a base from which conclusions can be drawn later.

7.1 Choice of anomaly detection approaches

There were many ML methods investigated during prestudy before choosing the final approaches. Notice that supervised and semi-supervised methods have been excluded in the thesis to enable real-world application where labels for every position of the UE, as well as the size of each cell, cannot be predefined and adjusted for every possible scenario.

The final approaches to use for evaluation were clustering- and density-based consisting of DBSCAN and LOF based on their performance. Distance-based with KNN were added for comparison, and its application for classification. Statistical- and isolation-based approaches was not suitable and will be elaborated.

7.1.1 Statistical-based

By simulating an ideal environment and for example not consider signal obstructions or smooth transitions between the beam areas in the hexagon cell, the data will not follow a statistical distribution. This approach is also assuming feature independence, used in histograms for example, which is not suitable for high dimensional data. Our data could not be depended on a single parameter and a statistical based approach is therefore irrelevant.

7.1.2 Isolation-based

Even though iForest at first seemed promising, it could not be implemented with the given dataset. The distribution of data points in the beams were skewed, as seen in Figure 5.9. The ratio of samples between two wide beams could for example be 1:45 for a specific random scenario. The procedure of iForest is based on training on batches, resulting in a tougher challenge to distinguish the clusters with fewer data points in total. iForest did struggle differentiating real inliers and outliers and was therefore not relevant for our problem.

7.2 Evaluation of ML methods

DBSCAN yielded the highest and most visually accurate number of outliers, followed by LOF and lastly KNN. Its precision and recall were also significantly better than the other two methods. DBSCAN and LOF can distinguish similar outliers due to their common concepts, but DBSCAN seemed to capture more subtle outliers, i.e., points that lied slightly outside the main lobe. However, as will be discussed in the Section 7.5, it is at this moment not possible to determine exactly which points that should be classified as outliers. This is valid at least when looking at the borderlines of the clusters. We did for example discover that some samples at the borderline could be misclasified by DBSCAN due to a higher number of neighbors, within ϵ, of the incorrect wide beam.

Applying the performance measurements to our problem, precision is considered to be more important than recall, thus desired to be high. We would rather have outliers classified as inliers than inliers classified as outliers. If our model would miss a side lobe, thus classifying it as a main lobe, the result would be as it is in the current product. That is, the result would not be better or worse than as it currently is. If a UE is connected to a main lobe and our method classifies it as a side lobe instead, a connection from a good beam to an inefficient beam would be made, hence yielding a worse connection as well as unnecessary beam switches. Hence, by trying to make sure that the data we classify as side lobes actually are side lobes, we make sure that the result is only improved. Weighting precision higher than recall resulted, as seen under F-score, can be an indication of what method is preferable. DBSCAN has a significantly higher F-score than the remaining methods. Even if KNN has a high precision, its F-score is much lower than LOF and therefore is LOF still considered as a better alternative than KNN.

7.3 Evaluation of performance measures on different data

As one can see in Table 6.3, the precision is high for all the tested sets when using DBSCAN. Recall however varies due to nearby clusters that can be considered as borderline cases. Based on the function of DBSCAN, these samples are not considered as outliers due to their fulfillment of number of neighbors within ϵ. If there would be a strict threshold for the transition of wide beam, they would have been considered as outliers and therefore decreased the recall dramatically. A strict threshold is not relevant in our case and have therefore affected the recall negatively.

With the same reasoning as above when applying the performance measurements, a high precision is preferred. The method has a high precision for all the different data sets and in general a high F-score as well. DBSCAN seem to be able to handle different datasets.

7.4 Distribution of outliers

In Figure 6.8, one could see that the number of outliers belonging to wide beam 5 and 8 were significantly higher than the remaining ones. From 5.7, these wide beams represents the ones propagating in a straight direction from the BS, where it seems like the side lobes are more likely to interfere constructively with its main lobe and thus a slightly better RSRP than the actual main lobe in the area. As one could see in Figure 5.7, the farther from the antenna the broader coverage area, which is also a reasonable explanation for the higher number of outliers belonging to wide beam 8 compared to wide beam 5 as shown in Figure 6.8. That the same figure and explanation does not seem to be valid for the other wide beams can be due to the variation of the number of outliers depending on the seed. With respect to wide beam 5 and 8, the occurrences of outliers from the others are significantly lower.

In Figure 6.7, one can also see that the outliers are mostly distributed in a diagonal direction from BS, outliers mainly belonging to wide beam 5 and 8. Similar diagonal pattern, angled with respect to each wide beam, for outliers belonging to the other wide beams could be noticed but not confirmed due to the lack of outliers. A possible explanation here as well is that they have interfered constructively, hence the slightly better wide beam RSRP. The difference was in general roughly 1% as presented in Section 6.5.1.

7.5 Limitations

Since it does not exist any baseline for which data points that belong to a side lobe or not, there is no fully accurate way to evaluate the methods that are used. An artificial baseline had to be created by using features that would not be available outside of the simulator, together with the ML method deemed to give the most correct results. More specifically, the features that were used for this purpose was the space coordinates (x- and y-coordinates) together with the beam indices. When plotting these in 3D one could easily distinguish outliers from inliers by only looking at the plots as seen in Figure 5.2, which was used when trying to decide which ML method that worked best. As can be seen in the same figure, there are many outliers that are clearly visible to the eye which could be exploited when deciding upon an ML method to use for the creation of the baseline. However, due to human error, the performance measures will not be completely accurate when later trying to decide upon an ML method to be used as the classifier. The ordering of the performances will probably be correct, i.e. the one ML method that is better than the others. However, the actual value of the performance measures will probably be somewhat lower when used in reality.

In reality and with reflections, the data is not as clean as it is with LOS. The data points would probably be more scattered and DBSCAN would have a harder time to identify and classify outliers correctly. Challenges adjusting the parameters would occur and DBSCAN would maybe not be as suitable as it is in LOS.

Another limitation with DBSCAN is the need of many data points to make sufficiently dense clusters among the wide beams closest to the BS. As one can

see in the figures using AoA and distance, such as Figure 6.2, the clusters closest to BS are containing a low number of samples that also are being scattered. For the method to work properly, enough of neighbors within the distance ϵ is required. DBSCAN works in LOS but can be questioned when expanding for further applications more similar to reality.

7.6 Beam update based on prediction

As one can see in Figure 6.7, the number of outliers is decreased significantly if choosing the upcoming top wide beam indices with the next highest RSRP.

7.6.1 Wide beam switches

There was an overall decrease in wide beam switches, although the change in number of switches varied depending on the number of outliers in each coverage area as well as the data set. We can see that the decrease in number of switches varies but always decrease when the predicted side lobes are avoided, which is what we wanted to achieve since it saves control signalling overhead. One can, however, see that there in dataset 5 is a much smaller decrease compared to the other datasets. The reason is mainly due to the occurences of many borderline cases, which our model had categorized incorrectly as inliers.

As have been mentioned throughout the report, a fewer number of beam switches can reduce the total energy usage. Even though the wide beam RSRP is decreased, the narrow beam RSRP and thus the TP did in general not. Performing a beam switch for those samples classified as outliers did result in both higher TP and lowered energy consumption, which is exactly what we had hoped for.

There were problems encountered when trying to extract the coverage area for each wide beam. The method used was to cut out rectangular-shaped areas in the area corresponding to a wide beam, adding a human source of error due to eye measurements. The areas were kept the same for all the datasets where borderline samples were clearly distinguishable. Bear in mind that the method to control the change in beam switches can be vastly improved, and has here just been used to show an improvement.

7.6.2 Wide beam and narrow beam RSRP

As one can see in Table 6.4, some changes in wide beam resulted in better narrow beam RSRP while other a lower instead. Most of the outliers resulted in higher narrow beam RSRP but we noticed that the variation could depend on where the outlier existed. Those outliers that were positioned at the borderline between two wide beams would normally yield a lower RSRP for the new chosen narrow beam. In this specific case, there was not a need of a beam switch even though the model predicted it would be preferable. An explanation to why the sample still was considered as an outlier is that it was not surrounded by enough of neighboring samples to be classified as an inlier, emphasizing the limitation of our baseline algorithm.

In general, the outliers were more accurate if they had been incorrectly connected to any of the wide beams considered in the middle of the hexagon cell, coincidentally also the wide beams with the most outliers as seen in Figure 6.8. These were less likely to be threshold boundary outliers.

Furthermore, we were able to confirm our hypothesis that avoiding outlier wide beams would result in better RSRP for the narrow beam and thus the connectivity and TP. We could also notice that the average 1.3 % drop in wide beam RSRP was mostly insignifcant, especially for those outliers that needed one beam refinement only. It is not surprising that the difference in percentage is relatively low for wide beam RSRP, due to the connection to a side lobe, and is a reason for only few occurences of outliers in the total data set. The difference between the narrow beam RSRP could, however, be twice as high as for the difference in wide beams, indicating a preferable choice. Due to the construction of the simulator, this information is not known and thus is the UE assigned to a narrow beam with lower RSRP. Even though a difference of a few percentages is relatively low, one needs to keep in mind that RSRP is measured in dB. Since dB is measured in a logarithmic scale, a few percentages of improvement still has a great impact of the total result. A change of 1 dB for example, changes the effect with a difference of 26% while a change of 3 dB can yield a difference of 100%. In our data, 1% is roughly 1 dB just as 3% is roughly 3 dB. Hence, even though we loose around 26% in effect in wide beam RSRP, the increase in effect from the narrow beam RSRP on about 100% is substantially greater.

7.6.3 Throughput

As we saw in Figure 6.12, the average TP was increased when using our ML algorithm to avoid side lobes. As mentioned in Section 6.5.3, shorter simulations of 5 seconds were performed for the remaining samples of the investigated ones for narrow beam RSRP. The three datasets in Figure 6.12 did however show noticeable fluctuations and were therefore investigated further. The fluctuations were considered being the result of the TCP regulation. The low TP in the data samples, connected to side lobes, was probably because of TCP detecting high latency in the data transfer, leading to TCP reducing tranfer speed. In other words, the unstable properties of the side lobes leads to the TCP reducing data transfer speed and hence TP.

As for the amount of beam switches, the results and differences in TP varied significantly although the trend is the same. That is, we always see an increase in TP when avoiding the side lobes.

7.7 The simulator

The process of generating the results to confirm improvements of the prediction when applying our model on simulated data turned out to be extremely time consuming and tedious, specifically for narrow beam RSRP. The results we were able to generate did confirm our hypothesis although we were not able to generate enough results to validate our results statistically with the time left for our project. This has therefore been left as future work.

7.8 Ethical aspects

One ethical aspect that would be needed to take into consideration is that our ML algorithm is based on an estimation of the locations of the UEs. Meaning that if a UE would want to connect to the network with our method incorporated in the product, it would have to let Ericsson use an estimation of their location which might be considered as some level of trespassing of the users privacy. Nevertheless, as we have elaborated on previously, it is unlikely that our algorithm would be applicable on real data and was more used to investigate the results of avoiding side lobes. However, we talk about incorporating the RSRP in the ML algorithm as future work and RSRP is as well an indirect estimation of the UEs location. There are many applications on the UEs today that uses the location based on GPS etc., the user is often asked whether they allow the application to use that information which would not be an applicable solution in our case.

In addition, one positive societal aspect is that a lot of energy could be saved if applying our algorithm due to the reduction of wide beam switches and hence control signalling overhead.

Chapter 8
Conclusion

In this chapter the results and discussion from previous chapters are put together and used to reach a conclusion coupled to the objective of this work.

8.1 Overall results

Of all the different ML methods that were investigated and evaluated, DBSCAN together with KNN as predictor gave the most promising results as they were able to detect the obvious outliers as well as the more subtle ones. This could be confirmed by visualization and it is therefore possible to find a ML method that performs in a satisfactory manner. Furthermore, by avoiding most of the predicted outliers, the number of beam switches was decreased significantly which improves control signalling overhead. There was an expected subtle average decrease in wide beam RSRP with 1.3 %, but on the other hand, an average increase in narrow beam RSRP with about 3 %. In addition we could confirm an increase in TP as well, although the increase could vary significantly. These results confirm that a greater signal connectivity can be achieved by avoiding side lobes by using ML for outlier detection. In other words, we were able to achieve all objectives of the thesis.

8.2 Future work

A subsequent project with the objective to improve BM further would be to go beyond LOS and find an ML method that detects beam reflections and still chooses the reflected beam if its strength still is considered as strong. Currently, if a main lobe is being reflected, its strength will decrease with the propagating distance leading to the UE being assigned to a side lobe. Nevertheless, these signals are also very unstable and strong only in very limited intervals in space, meaning it could improve the performance of BM by preventing UEs from being assigned to those side lobes.

Another project directly coupled to our work would be to find a way to use more features in the ML method to improve performance of our method further. Ours was limited to few dimensions due to the importance of visualization. An approach was to find a pattern between the signal strength between the 12 existing beams, where nearby beams had a significantly higher RSRP than those farther

away. Challenges related to curse of dimensionality and explanability did however occur when we investigated this further.

A baseline could be created where the trained data set also would contain which data points that are outliers, and which are not. One would by then no longer be dependent on visualizations to evaluate the ML methods used and could rely fully on evaluation measures.

References

[1] 3rd Generation Partnership Project (3GPP). *Technical Specification Group GSM/EDGE Radio Access Network; Radio subsystem synchronization.* TS 45.010, release 6, 2003.

[2] 3rd Generation Partnership Project (3GPP). *Technical Specification, 5G NR; Physical channels and modulation.* TS 38.211, release 15, 2018.

[3] 3rd Generation Partnership Project (3GPP). *Technical Specification, 5G;NR; 5G NR; Physical layer measurements.* TS 38.215, release 15, 2018.

[4] Bensky, Alan. *Short-range Wireless Communication.* 3rd edition. Elsevier. 2019.

[5] Barrera, Natalie. Ericsson. *How to make anomaly detection more accessible.* 2020. `https://www.ericsson.com/en/blog/2020/7/how-to-make-anomaly-detection-more-accessible`. *(Access 2021-04-20)*

[6] Barney, John. Ericsson. *The future of connected vehicles lies with 5G.* 2019. `https://www.ericsson.com/en/blog/2019/5/5g-role-in-future-of-connected-vehicles`. *(Access 2021-03-18)*

[7] Breunig, M. M., Kriegel, H. P., Ng, R. T., Sander, J. *LOF: Identifying Density-Based Local Outliers.* Institute for Computer Science, University of Munic. 2000. `https://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf` *(Access 2021-03-18)*

[8] Dahlman, E., Parkvall, S., Sköld J.. *5G NR: The Next Generation Wireless Access Technology.* Academic press. 2018.

[9] Ester, M., Kriegel, H. P., Sander, J., Xu, X.. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.* Institute for Computer Science, University of Munic. 1996. `https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf` *(Access 2021-03-18)*

[10] Federal Standard 1037C. *Telecommunications: Glossary of Telecommunication Terms* Institute for telecommunication sciences. `https://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm`

[11] Giordani, M., Polese, M., Roy, A., Castor, D., Zorzi, M.. *A Tutorial on Beam Management for3GPP NR at mmWave Frequencies*. Institute of Electrical and Electronics Engineers. 2019. https://arxiv.org/pdf/1804.01908.pdf *(Access 2021-03-18)*

[12] Liu, F.T., Ming Ting, K., Zhou, Z.H.. *Isolation-based Anomaly Detection*. Monash University, Nanjing University. https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/tkdd11.pdf *(Access 2021-03-18)*

[13] Patel, H. H., Prajapati, P.. *Study and Analysis of Decision Tree Based Classification Algorithms*. International Journal of Computer Sciences and Engineering. Vol.6, Issue-10, Oct. 2018

[14] Sasaki, Yutaka. *The truth of the F-measure*. University of Manchester, Manchester. 2007. https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf. *(Access 2021-04-20)*

[15] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. https://cs.nyu.edu/~roweis/papers/ncanips.pdf

[16] Shi, T., Horvath, S. *Unsupervised Learning With Random Forest Predictors*. Journal of Computational and Graphical Statistics, Vol. 15, No. 1, Pages 118–138. 2006.

[17] Togbe, M., Barry, M., Boly, A., Chabchoub, Y., Chiky, R., et al.. *Anomaly De-tection for Data Streams Based on Isolation Forest using Scikit-multiflow.*. The 20th International Conference on Computational Science and its Applications (ICCSA 2020). 2020. Italy. https://hal.archives-ouvertes.fr/hal-02874869/document *(Access 2021-03-18)*

[18] Wang, Boqiang. *Adaptive Beam Management in 5G-NR: A Machine Learning Perspective*. Lund: Department of Electrical and Information Technology, Lund University. 2020

[19] Xu, X., Liu, H., Li, L., Yao, M. *Comparison of Outlier Detection Techniques for High-Dimensional Data*. International Journal of Computational Intelligence Systems, Vol. 11 (2018), 652-662